Semi-Supervised Bio-Named Entity Recognition with Word-Codebook Learning

Pavel P. Kuksa[*] Yanjun Qi[†]

Abstract

We describe a novel semi-supervised method called Word-Codebook Learning (WCL), and apply it to the task of bio-named entity recognition (bioNER). Typical bioNER systems can be seen as tasks of assigning labels to words in bio-literature text. To improve supervised tagging, WCL learns a class of word-level feature embeddings to capture word semantic meanings or word label patterns from a large unlabeled corpus. Words are then clustered according to their embedding vectors through a vector quantization step, where each word is assigned into one of the codewords in a codebook. Finally codewords are treated as new word attributes and are added for entity labeling. Two types of word-codebook learning are proposed: (1) General WCL, where an unsupervised method uses contextual semantic similarity of words to learn accurate word representations; (2) Task-oriented WCL, where for every word a semi-supervised method learns target-class label patterns from unlabeled data using supervised signals from trained bioNER model. Without the need for complex linguistic features, we demonstrate utility of WCL on the BioCreativeII gene name recognition competition data, where WCL yields state-of-the-art performance and shows great improvements over supervised baselines and semi-supervised counter peers.

Keywords: semi-supervised feature learning, information extraction, sequence labeling, biomedical natural language processing, named entity recognition

1 Introduction

For biomedical research, efficient access to information contained in online scientific literature collections is essential as it plays a crucial role to the initial experiment planning, the final data interpretation, or communication of results. Most of the biomedical discoveries are communicated through publications, or reports. Recently, biomedical databases enabled efficient data retrieval, exchange, and analysis of previous discoveries. Most of these databases rely on manual annotations of the literature through field experts, which is often associated with significant curation workloads [20], and limited to a fraction of journals.

The life science literature database PUBMED stores around 16 million citations and receives over 70 million queries every month [19]. The rapid accumulation of new publications that appear on a daily basis needs to be processed in time to extract new discoveries and/or to revise existing ones. A range of text mining and natural language processing (NLP) strategies has been proposed to convert human language in bio-literature into formal computer representations for sophisticated biomedical literature mining.

Finding biomedical named entities is one of the most important subtasks in processing bio-literature text. Technically speaking, finding bio-named entities in bio-literature is similar to common named entity recognition (NER) systems which label atomic elements in newspaper text into categories such as person name, company name, or organization name [10]. Bio-named entity recognition is critical for subsequent tasks like document retrieval, information summarization, or bio-event extractions.

The task of bio-named entity recognition (bioNER) differs from other common NER problems in several aspects, which make bioNER substantially more difficult:

- Much more rare words, extensive usage of acronyms and constantly changing vocabulary. For instance, there exist millions of gene names used. Rare word problem results in the need of being able to detect and predict previously unseen words/entities.
- Complex orthographic patterns, many variations of bio-terms referring to same entities. This substantially expands active vocabulary, and complicates building named entity dictionaries.
- Multiple types of bio-named entities exist with similar morphology and contexts, such as protein name, gene name, disease name, cell name, etc. For example, gene names naturally co-occur with other bio-named entity types, such as cell names.
- Ambiguity. The same name may refer to a range of biological objects and terms, which further complicates their distinctions.

Typical bioNER systems assign labels to words in biomedical literature. Supervised techniques have yielded great success, though the performance is restricted by the expense of annotating the corpus [29]. Recently semi-supervised learning has become one of the most natural forms of training for language processing tasks, since unlabeled language data is plentiful.

In this paper, we propose a semi-supervised method called Word-Codebook Learning (WCL), with the hope of tackling/easing above-mentioned issues through the usage of large unlabeled corpora. The basic motivation is the following: words are basic units of meaning and fundamental building blocks of human language. For bioNER task, individual words carry significant semantic and label information. WCL aims to learn a class of

[*]Department of Computer Science, Rutgers University.
[†]Machine Learning Department, NEC Labs America Inc.

word-level feature representation (embeddings) to capture multiple indications of individual words through the usage of unlabeled bio-literature sentences. Two kinds of WCL are proposed: (1) General WCL, where an unsupervised method based on neural networks tries to use contextual similarity of words to learn accurate word representations describing semantic information; (2) Task-oriented WCL, where a semi-supervised method tries to capture each word's target label patterns from unlabeled data using supervised signals from iteratively trained bioNER model. After getting embedded feature representations, words are clustered according to these vector embeddings. A vector quantization (VQ) technique is utilized to assign each word to one of the codewords in a codebook learned through VQ. Finally codewords are treated as extra word *attributes* and added into bioNER systems for named entity labeling.

We apply WCL on the BioCreativeII gene name recognition competition [29] data set. Our methods are compared to several baselines, including the fully-supervised learning, the self-training, the semi-supervised counter peer, and a simple co-occurrence based WCL. With all settings we tried, WCL shows significant improvements and yields the state-of-the-art performance (F1 test score **87.86**) without the need for complex linguistic features and domain dictionaries.

2 Related Work

2.1 Supervised Sequence Labeling for Natural Languages BioNER belongs to the general family of information extraction (IE) tasks in the natural language processing (NLP) field. NLP research aims to convert human language into representations that are easy for computers to process. Typical problems in NLP range from the syntactic, like part-of-speech tagging, chunking, to the semantic, such as named entity extraction (NER) [7]. Most of these sub-tasks could be treated as sequence segmentation or sequence labeling problems. For instance, NER systems label atomic elements of the sentence into several categories, which essentially classifies words into one of the multiple classes. While early studies were mostly based on handcrafted rules, recent systems use supervised machine learning techniques to automatically train labeling algorithms from a collection of training examples. Popular supervised techniques includes Hidden Markov Models (HMM), Maximum Entropy (ME) Models, Support Vector Machines (SVM), and Conditional Random Fields (CRF). See [24] for a review of NER problems.

2.2 Semi-supervised Learning It is quite difficult to obtain labeled training sentences with word-level annotations, especially for bio-literature text. Supervised NLP techniques are restricted by the availability of labeled examples. Semi-supervised learning has become one of the most natural forms of training, since unlabeled language data is abundant in this field. For instance, PUBMED (the central life science publication database) provides free downloads for all its publication abstracts (more than 16 million citations with over ~1.3G tokens after preprocessing).

Self-training [27, 15, 18], and co-training [2, 5] are popular semi-supervised methods applied in NLP. Both utilize large sets of unlabeled corpora and try to improve over supervised methods by iteratively adding self-labeled *examples* predicted by the current model. This can give improvements to a model, but care must be taken as the predictions are prone to noise. Many other semi-supervised learning algorithms exist, including Transductive SVMs [14, 6], graph-based regularization [32], entropy regularization [12] and EM with generative mixture models [25], see [4] for a review.

Unlike most popular semi-supervised approaches, the proposed WCL method induce word *features* (codewords, see Section 3.2) (not *examples*) from a large corpus of unannotated examples. These features are then used to augment the feature space of the labeled set. This is an orthogonal direction for improving accuracy, i.e. it can be used in combination with many other supervised or semi-supervised methods.

2.3 Bio-Named Entity Recognition (bioNER) and Gene Name Finding A considerable fraction of the existing discoveries in life science are in the form of natural language text. Currently most functional information contained in biological databases has been extracted directly or indirectly from these text sentences. Many previous efforts have been made to extract biologically relevant information from electronic biomedical texts automatically. See [20] for a survey.

While complete automatic semantic understanding is still a far-distant goal, researchers have taken a divide and conquer approach and identified several important sub-tasks such as bio-entity recognition, protein interaction extraction, etc. When processing bio-literature text, bioNER is one of the most important tasks since biological entities represent the primary targets of scientific discoveries. A combination of characteristics makes bio-name entities difficult to recognize automatically (details in Section 1). Well-labeled training sentences are scarce for general bioNER types, with GENIA [16] corpus as one of the first attempts.

Among typical bio-entity types, gene/protein names are among the most fundamental ones since genes/proteins are the chief targets for biomedical science. The problem of finding gene names (termed as "gene mention" (GM)) has been one of the major tasks in the challenge competition series: BioCreativeI and BioCreativeII [19]. Two BioCreative competitions provided well-defined GM corpora and the state-of-the-art evaluation results to compare on this benchmark data set [29]. Nineteen teams participated in BioCreative II gene mention task and each team submitted one to three runs. All top teams utilized large domain lexicons (termed "gazetteer"), complicated linguistic features (such as POS), and complex training strategies (for instance, two passes training of sequence models, forward pass + backward pass).

2.4 NER with Words Grouping WCL essentially clusters words in the vocabulary into multiple groups. Several previous works [3, 23] made efforts in this direction to solve NER tagging. For example, Miller et al. [23] proposed to augment annotated training data with hierarchical word clusters that are automatically derived from a large unannotated corpus. The words were grouped according to their co-occurrence statistics (with method proposed in [3]) in the unlabeled sets. Since co-occurrence patterns could be treated as embedding and fit into WCL as well, we make experimental comparisons of the two proposed WCL (General or Task-Oriented) methods to a co-occurrence based WCL baseline in Section 5.

Another group of closely related methods treat word clusters as hidden variables in their models. For instance, Koo and Collins [17] proposed a conditional log-linear model, with hidden variables representing the assignment of atomic items to word clusters or word senses. The model learns to automatically make the cluster assignments based on a discriminative training criterion. The method was tested within reranking problems.

3 Methods

In the following, we describe the word-codebook learning (WCL), which has two steps: (I) learning word-level feature representations (embedding vectors), either with respect to semantic meaning (unsupervised) or target label patterns (semi-supervised), Section 3.1; (II) building word-codebook and assigning codewords to every word according to its embedding vector (Section 3.2).

The algorithm of general word codebook learning is given in Table 1 and is further illustrated in Figure 1. Embedding step (step 3 in Table 1) is accomplished with suitable learning procedures (details in Section 3.1). Word codebook construction and codeword assignment (steps 4 and 5 in Table 1) are described in Section 3.2.

3.1 Step I: Learning Word-Level Embedding

WCL relies on the key observation that individual words carry significant semantic information or label

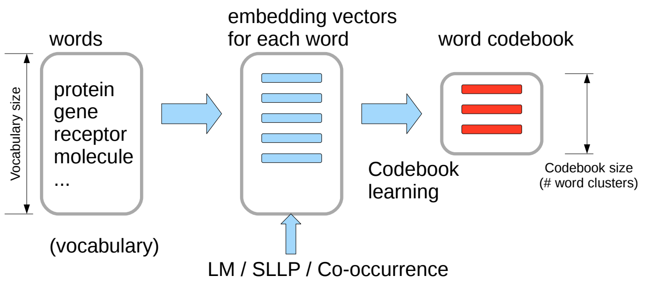

Figure 1: Flow chart of word-codebook learning (WCL)

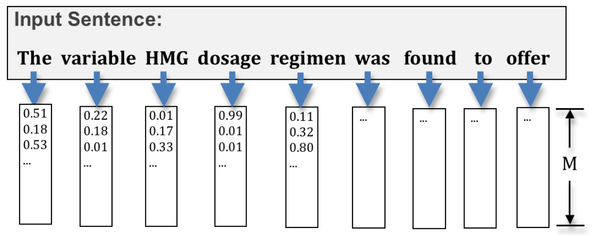

Figure 2: Word embedding step maps each word to a vector of real values (with dimension M) which are learned from the large unlabeled data set (e.g., PubMed abstracts for bioNER).

indications for NLP tasks. Thus we try to map each word to a vector of real values (called "embedding" in the following) which is able to describe this word's semantic meanings or possible class patterns. Figure 2 illustrates this mapping step with an exemplar sentence. We propose two strategies to learn embeddings which represent every word's

- Semantic information: an unsupervised approach named "language model" (LM) to capture words' semantic meanings in human language (e.g., bio-literature text here). This strategy makes use of only unlabeled set and is not related to the targeted class types. Thus we name WCL using this strategy as "General WCL".
- Class pattern: a semi-supervised method named "self learned label patterns" (SLLP) to capture words' target class (e.g., named entity types) distributions. This strategy depends on the target class types WCL is targeting. Thus WCL using this strategy is named as "Task-Oriented WCL".

3.1.1 General Embedding: Language Model (LM)

Traditional language models estimate the probability of the next word being w in a language sequence. Collobert et al. [7] proposed a different type of "language model" (LM) which is an unsupervised task to embed normal English words into a M dimensional space by utilizing unlabeled sentences from Wikipedia. We adapt this approach to bio-literature text and train

Algorithm for Word Codebook Learning (WCL)
1: Define the feature representation $\phi(w)$ for words w
2: Choose a target word vocabulary $\mathcal{D}$
3: Train $\phi(w)$ for all words in the word vocabulary to obtain a collection of word vector embeddings
4: Form a codebook $\mathcal{C}$ from the vector embeddings to obtain a word codebook
5: Augment the representation of words $\phi(w)$ with their learned codebook feature

Table 1: Word codebook learning (WCL)

the language model on PubMed abstracts. The basic motivation is that in most language tagging tasks, words semantically similar can be usually exchanged with no impact on the labeling. For example, in a sentence like "EGFR interacts with an inhibitor" one can replace "interacts" with "cooperates" with no change in the sentence meaning.

The proposed LM tries to force two natural sentences with same semantic tags to have similar embedding representations and force two sequences with different semantic tags to have different representations. This is achieved by utilizing unlabeled text fragments and learning to predict whether the given text sequence exists naturally in bio-literature, or not. Figure 3 gives a schematic framework of LM embedding.

Multiple-layer perceptron network (one type of neural network (NN)) is used for the LM embedding learning. The first layer is a lookup-table layer which map raw words into real-valued vectors (which are the embedding vectors we are looking for). With a sliding window approach (where the inputs are word windows of a fixed size, and the middle word in the window is the word to be tagged), values of words in the current text window are concatenated and fed into subsequence layers which are classical neural network (NN) layers. The weights of all the layers in the network are automatically trained by backpropagation. The last layer outputs a scalar value $f(\mathbf{x})$, for each input $\mathbf{x}$ (a window of words). The model is trained with a ranking-type cost:

$$\sum_{\mathbf{x}\in\mathcal{X}} \sum_{w\in\mathcal{D}} \max\left(0,\, 1 - f(\mathbf{x}^{+}) + f(\mathbf{x}^{-})\right) , \tag{3.1}$$

where $\mathcal{X}$ is the set of all possible word windows in natural text, $\mathcal{D}$ is the vocabulary of words, and $f(\cdot)$ represents the output of NN architecture. $\mathbf{x}^{+}$ represents real text fragments (positive example) and $\mathbf{x}^{-}$ is a pseudo negative example which is generated from $\mathbf{x}^{+}$ by replacing its middle word w by a random word. We sample this cost *online* w.r.t. $(\mathbf{x}, w)$. With the above cost, LM is trained to recognize if the word in the middle of an input text window is naturally related to its surrounding words or not. Thus the learnt embedding vectors in the lookup-table layer are close for semantically similar words.

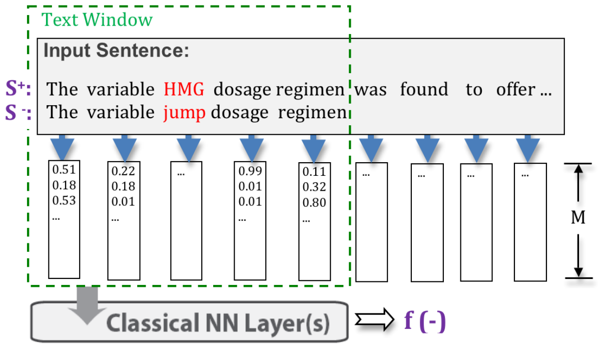

Figure 3: Word embedding with "language model" using a neural network (NN) framework.

3.1.2 Task Oriented Embedding: Semi-supervised Self-Learned Label Patterns (SLLP)

We now discuss a task-oriented method for learning word representations. The method directly targets a labeling task at hand by learning each word's label distribution patterns which represent probabilities with which a word might belong to the classes defined by the task.

Figure 4 gives a schematic summary of this approach, the so called "self-learned label patterns" (SLLP) learning. The basic version of SLLP is presented in [26]. Here we propose several extended versions.

For a given word w, a feature vector $\text{SLLP}(w)$ models the probability of assigning each target class to this word. Assuming the target label has K classes total, $\text{SLLP}(w) = (\text{SLLP}(w)_1, \ldots, \text{SLLP}(w)_K)$ with the ith component defined as

$$\text{SLLP}(w)_i = P(\mathbf{y} = i | w, \text{ where } w = \mathbf{x}_m), \tag{3.2}$$

where $m = (|x| + 1)/2$. That is, the i^{th} dimension measures the probability of label $\mathbf{y} = i$ being assigned given word w is the middle word (word of interest) in the input segment $\mathbf{x}$ (Figure 4). Here we are also using a sliding window approach ($\mathbf{x}$ represents a text window of five words in Figure 4). This distribution is of course unknown but can be empirically estimated from the training set or, critically, can be *re-estimated* using an unlabeled corpus by applying a classifier trained on the labeled set and averaging predicted labels over all cases in the unlabeled corpus.

The empirical SLLP is then defined for a given word w as:

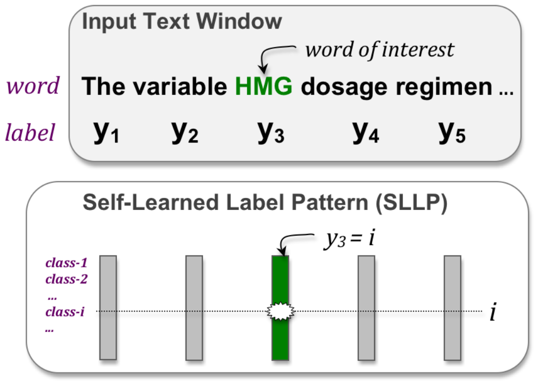
Input Text Window
word of interest
word
The variable HMG dosage regimen ...
label
y_1 y_2 y_3 y_4 y_5
Self-Learned Label Pattern (SLLP)
$y_3 = i$
class-1
class-2
...
class-i
...
i

Figure 4: Word embedding with SLLP (Self-Learned Label Pattern) learning.

$$\overline{\text{SLLP}}(w)_i = \frac{|\{j : f(\mathbf{x}_j^*) = i \ \wedge \ w = (\mathbf{x}_j^*)_m\}|}{|\{k : w = (\mathbf{x}_k^*)_m\}|}, \tag{3.3}$$

where $f(\cdot)$ is a bioNER classifier trained to predict label $\mathbf{y}$ given $\mathbf{x}$. Essentially this distribution measures the proportion of text sequences assigned to class i given word w is the word of interest in those window sequences.

As pointed out by the organizers of BioCreativeII GM task, it seems to be infeasible to detect 'difficult' (i.e. unique, rare) gene names, because they obey a Zipf-like distribution and there are as many unique and difficult gene names as there are common and easy ones [29]. In the following we try to conquer this difficulty two extended SLLP.

[CRKL], **an adapter protein** ...
[SH2-SH3-SH3] **adapter protein** ...
high **expression of** [p51a] ...
zygotic **expression of** [Kr] ...

Table 2: Exemplar words carrying rich class boundary information. Each row gives an example where the words next (or very close) to the named entity (gene name) are informative.

Boundary SLLP Rare words are normally the hardest examples to label since the training set could hardly cover them. In this case, we try to model those words which occur frequently before (or after) a certain class of entity type. During testing, when a rare word needs to be labeled, its neighborhood words might give strong indications of its target class types if these words always appear around the boundary of a certain entity class. Table 2 lists several exemplar words in bio-text that are usually very close to named entities (gene names). Clearly, some words carry strong class boundary information. In this case, basic SLLP can be extended to incorporate the class boundary distributions, named as "boundary SLLP".

Boundary SLLP (BSLLP) models how likely a word w is to occur before or after a certain class (assuming class t):

$$\begin{aligned} \text{bSLLP}(w)_{t,1} &= P(\mathbf{y} = t | w \in \{(\mathbf{x})_1, \ldots, (\mathbf{x})_{m-1}\}) \\ \text{bSLLP}(w)_{t,2} &= P(\mathbf{y} = t | w \in \{(\mathbf{x})_{m+1}, \ldots, (\mathbf{x})_{|\mathbf{x}|}\}) \end{aligned}$$

where m is the middle index in a given "window" $\mathbf{x}$ (sequence), as before. We show experimentally that this extension effectively improves the gene name recognition task which suffers a lot from the "rare word" problem.

Bigram SLLP While both the basic method and the boundary SLLP estimate word-class probability for individual class labels, to better capture context in which word appears, one could estimate distributions of target class label n-grams (n-*gram* SLLP) as opposed to distributions over individual class labels.

Here we consider a *bigram* SLLP model as an example. The Bigram-SLLP estimates for a given word w a set of probabilities using current tag (t_0), previous tag (t_{-1}), and next tag (t_1) tags (a simplified equation is provided to save space):

$$BiSLLP(w) = [P(t_{-1}, t_0|w), P(t_1, t_0|w)] \tag{3.4}$$

The BiSLLP feature vector, for instance, in IOBES representation has 25 dimensions corresponding to valid label bigrams (some of the combinations are invalid, e.g., IB, OI, etc.)

Compared to the basic/boundary SLLP that only looks at the individual label distribution independently for every word the n-gram SLLP estimates for every word the joint distribution over the label sequence corresponding to the labels of the word itself and the words surrounding it. Thus the n-gram SLLP gives more information about the potential labels for not only the word under consideration, but also for the words around it.

3.2 Step II: Building Word-Codebook

We now describe a vector quantization (VQ) step that is used to build the word codebook on words' embedding vectors, where every word is going to be assigned into one of the codewords (Figure 5).

For a given word w, the embedding step (e.g., SLLP, LM, etc.) defines a feature vector $\mathrm{E}(w) \in \mathbb{R}^M$. Similar feature vectors $E(w)$ indicate either semantic closeness of the words (as in LM) or closeness in terms of label assignments (as in SLLP). Thus grouping similar feature vectors together might give stronger indications of the target sequence labels and allow for better generalization over learned patterns.

Here we utilize a vector quantization technique [11] to convert embedding vectors $E(w)$ to prototype vectors. Each of the input vectors (embeddings to be quantized) would be quantized into one quantization level

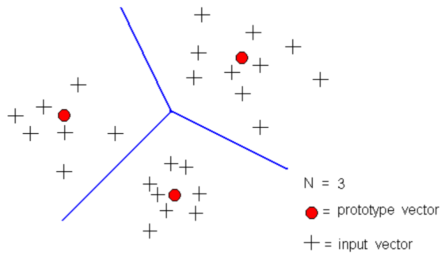

Figure 5: Building word-codebook step clusters words into multiple groups according to their vector embeddings. Each symbol "+" represents "input vector". Each red circle represents the "code vector". Blue boundary lines split the space into three regions and each input vector "+" would be assigned to the red circle in the corresponding region.

(called a *code vector*). The set of all *code vectors* is called a *codebook*.

Formally speaking, we have feature vectors $E(w)$ for every word w in the dictionary $\mathcal{D}$, i.e. $\{\mathrm{E}(w_1), \mathrm{E}(w_2), ..., \mathrm{E}(w_D)\}$, where $\mathrm{E}(w_d) \in \mathbb{R}^M$. Using $\mathcal{C}$ to represent the codebook set which includes N codebook vectors, $\mathcal{C} = \{C_1, C_2, ..., C_N\}$, VQ tries to minimize the following objective function,

$$\sum_{d=1}^{\mathcal{D}} ||\mathrm{E}(w_d) - C_n||^2, \text{ where } n \in \{1...N\} \tag{3.5}$$

With this optimization, VQ finds the best codebook $\mathcal{C}$ and for each input vector $\mathrm{E}(w_d)$, VQ finds the best code vector C_n to assign on. Figure 5 provides a schematic graph for this process. After VQ step, we use the indices of this code vector as the new feature for word w, called codewords. Essentially the whole process clusters all the words in the dictionary into multiple (N) groups and then uses the cluster ID as novel feature representation (a binary vector with all zeros except for the IDth dimension which is set to 1).

In summary, WCL learns embedded feature representations $E(w)$ for every word w from unlabeled examples (Section 3.1) and then uses the learned representations to build a word codebook $\mathcal{C}$ (Section 3.2). This, in turn, is used to re-represent word feature vector (Eq. 3.6). Assuming the original feature representation for word w in bioNER systems as $\phi(w)$, then

$$\bar{\phi}(w) = (\phi(w), CB(w)), \tag{3.6}$$

where $CB(w)$ is the codebook representation obtained from the unlabeled data (e.g., using LM or SLLP). WCL then re-trains the entity tagging classifier on the labeled set using this extended word feature representation. Figure 1 gives a flow chart of this process.

3.3 Advantages of WCL Our approach, similar to popular semi-supervised methods such as self-training, can be used as a wrapper approach with other supervised or semi-supervised base classifiers, to improve their prediction performance. In addition, WCL has the following merits:

- It learns compact representations (codebooks) for possibly very large number of words (e.g., PubMed has a word vocabulary of over 5 million words) through exemplar concepts (codewords). This allows to provide better generalization for labeling tasks, which otherwise need to work with and generalize over potentially infinitely many combinations of raw words.
- It allows to improve predictive performance by working with multi-resolution representations corresponding to various word groupings, from specific to general, which could better capture variety of natural classes/groups formed by words and therefore improve generalization capacity.
- It allows to improve predictive ability of simple models with very basic feature sets by adding well-trained WCL codebook features obtained using more complex, advanced models with rich feature sets.
- The WCL approach is highly scalable and can be used with large corpora as it adds features to the model without the need of adding large number of extra examples (e.g., as in self-training).
- WCL features give a robust abstraction of word groups, either semantic (as in general WCL with LM) or task-specific (as in WCL with SLLP) classes, to help improve generalization and labeling performance. The generalization is obtained through use of word clusters (codewords) instead of raw words, enabling better recognition of previously unseen combinations of words (or word's context/boundary) that are similar (in the sense of their WCL representation) to words in already seen examples.

4 Supervised & Semi-Supervised Baselines

Essentially WCL clusters words into multiple groups based on their semantic meanings or target-class patterns from unlabeled text. The constructed codebook assigns *pseudo-features* (codewords), to each word and adds them into the labeled set. From three perspectives, we compare WCL to three related baselines (1) Semi-supervision: Co-Occurrence pattern based WCL; (2) Semi-supervision: Self-training; (3) Supervision: Conditional random fields (CRF).

4.1 Semi-Supervised Baseline: Co-Occurrence based WCL Word co-occurrence-based models are typical in many NLP applications, such as text categorization or information retrieval. In NLP area, as mentioned in Section 2.4, researchers have tried to group words based on their co-occurrence statistics and then added into NER systems for performance improvements. As a direct baseline, we treat each word's cooccurrence patterns to all the other words as an embedding representation. Thus this embedding could be added into WCL similar to LM and SLLP embeddings.

In this paper, co-occurrence based model is used to estimate semantic relatedness of the words based on their spatial co-location. We estimate the word co-occurrence matrix using short text fragments (windows) extracted from large text corpus (e.g., PUBMED). Words with similar co-occurrence patterns can be clustered together to define word classes containing related words. We estimate each entry (w_i, w_j) in the cooccurrence matrix by counting the number of times word j cooccurred with word i within all the possible text windows in unlabeled set.

4.2 Semi-Supervised Baseline: Self-Training As already mentioned, self-training [27] (also called 'bootstrapping' in the traditional NLP field) [2] augment the training set with labeled examples from the unlabeled set which are predicted by the model itself. Self-training can be used in combination with any learning model and is highly scalable to the size of target problems. However, it is vulnerable to the incestuous training bias problem [28, 31], i.e. examples may be consistently mislabeled making the model even worse on the next iteration. To combat this, several authors have proposed schemes for only adding examples that meet a selection criterion [18, 28, 8], but these heuristic choices still might yield unreliable results.

We apply self-training to the same supervised baseline method to compare to the performance of WCL. There are numerous variants of self-training. We adopt the following weighting scheme: given L training examples, we choose L/R (R is a parameter to choose) unlabeled examples to add in the next round of training.

4.3 Supervised Baseline: CRF Conditional random fields (CRFs) [21] achieve state-of-the-art performance across a broad spectrum of sequence labeling tasks, including the gene mention task in the BioCreativeII competition [29]. Thus we choose CRF as our supervised baseline. Linear-chain CRFs are discriminative probabilistic models over observation sequences $\mathbf{x}$ and label sequences $\mathbf{y} = (y_1, ..., y_{|\mathbf{y}|})$, where $|\mathbf{x}| = |\mathbf{y}|$, and each label y_i has K different possible discrete values (multi-class). The conditional probability is defined as

$$p_\theta(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp(\sum_j \theta_j F_j(\mathbf{x}, \mathbf{y})) \tag{4.7}$$

where

$$F_j(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} f_j(\mathbf{x}, y_i, y_{i+1}, i)$$

and

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \exp(\sum_j \theta_j F_j(\mathbf{x}, \mathbf{y})).$$

The model is trained by maximizing the log-likelihood of the training data by gradient methods, which is a convex optimization problem [21]. A dynamic algorithm (the forward/backward strategy) is used to compute all the required probabilities $p_\theta(y_i, y_{i+1})$ for calculating the gradient of the likelihood. We used the CRF++ toolkit [1] to train CRF models.

Multiple semi-supervised strategies have been extended to CRF, such as entropy regularization [13] or generalization expectation [9, 22]. Considering these methods are either unstable [22], or not applicable to the large scale of unlabeled corpus we use in this paper, they are not compared here.

5 Experimental Results

We now present experimental results for labeling bio-literature texts. We show applications of our method in the semi-supervised, as well as transductive and supervised settings.

5.1 Datasets & Settings The BioCreativeII gene mention (GM) [29] competition data set involves a sequence tagging task which looks for gene/protein names in a bio-literature text corpus. The GM competition data set contains 15000 training sentences and 5000 test sentences. We use a collection with more than 4.5M PUBMED abstracts from 1994 to 2009 as our unlabeled corpus for semi-supervised learning. The size of the training/test/unlabeled sets is given in Table 3.

Since almost all the top teams in BioCreativeII GM competition [29] utilized conditional random field (CRF) model, we use conditional random fields (CRF++ tool [1]) as our base classifier, and apply WCL as a wrapper semi-supervised approach.

In CRF model we tried the following word feature sets:

- words and their capitalization attributes in a 5-word window surrounding the current word (words+caps feature set)
- words, their capitalization attributes, prefix, suffix (length up to 4), and the string (letter) pattern in a 5-word window surrounding the current word (extended word feature set)

Train	15000 sentences	426184 tokens
Test	5000 sentences	143391 tokens
Unlabeled	60M sentences	~1.3G tokens

Table 3: Size of the used data sets. Train/Test were from BioCreative II GM competition

Note we did not use any named-entity dictionaries, or linguistic features (POS, chunk types, etc.)

For general WCL, we train the language model (LM) on text windows that we extract from PUBMED. The neural network used one hidden layer in its classical NN part for LM training and sliding text windows of 11-words as inputs. LM learned embedding vectors for the most frequent 40K words in PUBMED (with 50 dimensions in the lookup table layer). These word representations are clustered to obtain 1K codebook entries. Each word in the dictionary is then mapped to one of the 1K codewords. We add this codeword feature to the corresponding feature set and re-train the supervised CRF with the labeled set.

For the task-oriented WCL in the semi-supervised setting we also use PUBMED as the unlabeled corpus and estimate the word-class label distributions (SLLP features) for top most frequent words in PUBMED, similar to the general WCL. The SLLP feature vectors are then clustered into 256 codeword patterns. Similarly, SLLP codeword features are used to augment the labeled data and we re-train the supervised classifier (CRF) on the extended feature set.

In addition, SLLP features could be estimated from train or test sets as well. We call these two cases transductive and supervised experiments where the task-oriented WCL using both train and test sets (transductive) or train set only (supervised) as the "pseudo-unlabeled" corpus to learn feature embeddings. That is, in the transductive setting we use SLLP word codebook features estimated on train and test sets only, and in the supervised setting we use SLLP word codebook features estimated on the train set alone.

For cooccurrence WCL, we compute cooccurrence frequencies for pairs of words in PUBMED abstracts. We use a word window of size 11 and count the number of times two words cooccur within the text window and one of them is the middle word of current text window). The word cooccurrence vectors are then clustered to obtain cooccurrence-based word codebook (2950 codewords). Similar to other cases, the learnt codeword is added in next round of training.

Methods are evaluated using F1 score (as well as precision and recall). For bioNER task the evaluation is performed on the phrase level, not individual word level, i.e. precise identification of the multi-word named entity boundaries is required for its recognition.

In the following, we compare general WCL and task-oriented WCL models to various baselines, including full supervision, semi-supervision, and simple cooccurrence WCL. We also compare our results to the state-of-the-art results previously reported in the literature on GM benchmark data set from BioCreative competition.

5.2 Performance Under general WCL (LM Codewords) In Table 4, we use F1 score to compare GM performance using WCL with the language model (LM) codebook to the supervised CRF baselines, when using different base feature sets (words only, words+caps, and extended word feature set).

Method	**Baseline(CRF)**	**+ WCL(LM)**
Words only	74.29	**83.39**
Words+Caps	82.02	**86.19**
All word features	86.34	**87.33**

Table 4: F1 score on the test set for BioCreativeII GM. General WCL with Language model added on CRF. WCL (LM) improve over the supervised CRF in all three settings.

Adding WCL (LM) codebook features provides improvements over supervised CRFs for all feature setups. The best performance obtained with WCL (LM) plus basic word attributes (word capitalization, prefix/suffix, string appearance pattern) (F1 87.33) is slightly higher than the best performance reported in the BioCreativeII competition (F1 87.21). All the top systems in the competition used many other features, such as POS, added with extensive domain-specific dictionaries, and utilized complex techniques such as bidirectional training and multi-classifier output integration [29]. To summarize, general WCL using LM effectively improves GM performance and achieves state-of-the-art performance with only simple word features.

5.2.1 Impact of the codebook size We vary codebook size (N) from small (512 codewords) to large (4096 words) and get a range of performance impacts from changing codebook resolutions. We can see the impact of the codebook size on the performance in Table 5. The observed performance (F1 score) varies within 0.6 range from 85.48 to 86.19. We note that using multiple codebook features corresponding to several codebooks with different sizes can obviate the need to select a particular fixed size for the codebook, and may also provide better generalizations. For example, using two codebooks of size 512 and 1024 achieves F1 score of 86.26, slightly higher than results of individual codebooks. Here the baseline is CRF with only word and caps features (F1 82.02).

Codebook size	512	1024	2048	4096
WCL (LM)	86.16	86.19	85.64	85.48

Table 5: Impact of the codebook size on the performance (F1 score). Baseline uses CRF on Words+Caps which achieves F1 test score: 82.02.

5.3 Performance Under Task-oriented WCL (SLLP) Table 6 compares GM performances under multiple setups using WCL (SLLP). We could see that WCL with basic SLLP and WCL with boundary SLLP improve over the CRF supervised baselines. WCL with Bigram SLLP shows better improvements compared to basic and boundary SLLP.

Method	**Baseline (CRF)**	**+ WCL(SLLP)**
Words+Caps	82.02	**84.01** (basic)
Words+Caps	82.02	**85.24** (boundary)
Words+Caps	82.02	**86.12** (bigram)
All Features	86.34	**87.16** (boundary)

Table 6: F1 score on the test set for BioCreativeII GM. Clustered SLLP and Boundary SLLP added to CRF. Both improve over the supervised CRF when using only words or all word attributes.

The best performance achieved here (F1 87.16 in the last row of Table 6) is only slightly lower than the best team (87.21 F1 test) in the BioCreativeII GM competition. In summary, WCL with self-supervision (SLLP) could improve CRF on this GM corpus effectively and we could achieve state-of-the-art performance using only basic word features (which is similar to the general WCL with LM).

We also test WCL in the transductive setting (Table 7) using train/test set for SLLP codebook learning. We observe that even when SLLP codebook is estimated on the relatively small set (train+test), WCL could still provides effective improvements over the supervised CRF. These improvements, however, are much smaller compared to the improvements in the semi-supervised setting (Table 6) which used much large PubMed to learn. When only the train set is used with WCL, WCL could still improve to F1 83.13 compared to F1 82.02 of the supervised CRF.

Please note that WCL (SLLP) could be run iteratively, where SLLP are built through iteratively retrained CRF whose feature set is augmented with WCL (SLLP) from previous rounds. In Table 7, we handle the transductive case using two iterations. We found a small improvement exists from iteration 1 to iteration 2.

5.4 Performance under Multiple WCL Now we compare performance of the WCL with single codebooks

Method	**Baseline (CRF)**	**+ WCL(SLLP)**
Words+Caps	82.02	**84.65** (boundary, 1 iter.)
Words+Caps	82.02	**84.73** (boundary, 2 iter.)

Table 7: F1 score on the test set for BioCreativeII GM. Transductive setting. Clustered SLLP added to CRF. SLLP provides effective improvements in the transductive settings.

to the WCL which uses multiple codebooks. To obtain multiple codebooks, we combine WCL codebook features obtained using unsupervised WCL with LM, and WCL with semi-supervision (SLLP). Table 8 gives results for WCL using multiple codebooks. For both basic and extended word feature sets, using multiple codebook WCL improves over WCL with single-codebook. The best performance (F1 87.86) is achieved using basic word features (capitalization attributes, prefix/suffix, and the string pattern).

In Figure 6, we show distributions of the number of errors over the gene name lengths (number of tokens in each named entity) and compare the supervised CRF, the WCL with LM, and the multiple WCL. Multiple WCL results in the largest improvements in accuracy for single-word gene names and multi-word gene named entities with the length of up to 5 words. Using general WCL with LM, per-word error rates improve by about 18% compared to the supervised CRF. WCL with SLLP reduces word-level error by 16.6%. Using multiple WCL results in the 20% improvement in the error rate. WCL reduces the number of false negative (FN) gene named entities by around 26-35% for LM, SLLP, and multiple codebook settings, while reducing the number of false positives (FP) by 10-18% at the same time (Table 9).

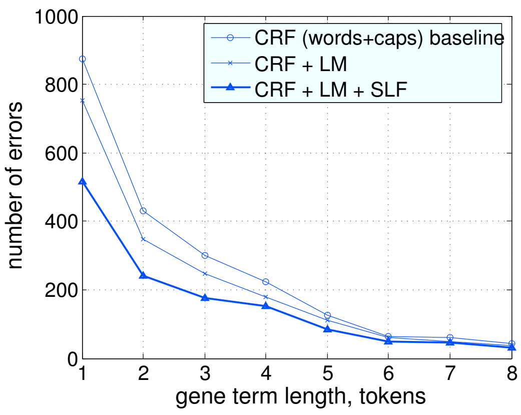

Figure 6: Distribution of errors per term (gene name) length. Both SLLP and LM improve over the supervised CRF.

Method	Baseline (CRF)	WCL (LM)	WCL (SLLP)	multiple WCL
Words+Caps	82.02	86.19	86.12	**87.16**
All word features	86.34	87.33	87.16	**87.86**

Table 8: F1 score on the test set for BioCreativeII GM. WCL with multiple codebooks improves over the supervised CRF and single-codebook setups.

Method	TP	FP	FN
Baseline CRF (words+caps)	4870	674	1461
+ WCL (LM)	5257	610	1074
+ WCL (SLLP)	5250	611	1081
+ multiple WCL (LM+SLLP)	5343	586	988

Table 9: Reduction in error for GM task. TP is the the number of true positive gene named entities; FP is the number of false positive named entities; FN is the number of false negative named entities. WCL improves over the supervised CRF.

Method	Baseline (CRF)	+WCL (SLLP)	+WCL (SLLP best)
Words+Caps	82.02	84.01(basic)	**84.98**(basic)
Words+Caps	82.02	85.24(bound.)	**86.39**(bound.)
Words+Caps	82.02	86.12(bigram)	**86.56**(bigram)

Table 10: F1 score on the BioCreativeII GM competition task. Improving basic models with better word codebooks. SLLP best in the last column refers to the use of SLLP codebook obtained from the best model with all word features. (bound. means boundary).

5.5 Improving Simple Models with WCL Codebooks from Better Models We now show that simple models with basic word features could achieve much better performance by using WCL codebook derived from better models that use more extensive sets of features.

In this experiment, we add WCL codebook feature obtained from the model with extended word feature set (words, caps, prefix/suffix, string patterns) to the model with the basic word feature (words+caps only) set. Table 10 gives the results for WCL with basic SLLP, boundary SLLP, and the bigram SLLP. We could see that adding WCL codebook feature from better models improves over using WCL codebook features from the basic models. For instance, using the pre-trained SLLP codebook from the model with extended word feature set (i.e. with the prefix/suffix and string pattern features), the model with only basic words+caps features achieves F1 score of 86.56, which compares well with the performance of the supervised CRF with all word features (F1 86.34) as well as with the WCL with SLLP and all word features (F1 87.16). In an important case of online named entity recognition systems, for which speed and efficiency are critical, training codebooks beforehand using models with extensive feature setups, and utilizing pre-trained codebooks with the simple models using only basic word features, could be an attractive choice for improving tagging speed while maintaining its predictive performance.

5.6 Comparison with Baseline: Co-Occurrence based WCL In Table 11 we show GM performance when we add co-occurrence WCL features to the CRF with the basic feature set. The observed improvement, though significant, is much lower when compared to WCL with LM or SLLP methods. For instance, WCL with LM achieves F1 score of 86.19, while F1 score of the co-occurrence-based model is only 83.72.

Method	Precision	Recall	F1
Baseline CRF (Words+Caps)	87.84	76.92	82.02
Baseline CRF (Words+Caps) + Co-occurrence	88.52	79.42	83.72

Table 11: Classification performance on BioCreativeII GM task for co-occurrence based WCL.

5.7 Comparison with Self-training We apply self-training to the baseline CRF to compare its performance to WCL. Table 12 gives results for the CRF with self-training on basic and extended word feature sets. Self-training improves over the baseline (supervised CRF) only in the case when all word features were used. We note the observed improvement with self-training is marginal, and is much smaller when compared to the improvements of WCL (e.g., Table 6).

Method	Baseline (CRF)	+ Self-training
Words+Caps	**82.02**	81.99
All word features	86.34	**86.43**

Table 12: F1 score on the test set for BioCreativeII GM. with Self-training applied on baseline CRF.

5.8 Comparison with Previous Results Previous best systems from the literature use complex techniques such as, for instance, bi-directional CRF models and multi-classifier settings with integration of outputs from several classifiers, combined with extensive usages of domain dictionaries/lexicons, to achieve good performance. As we can see from the comparison with the semi-supervised and supervised baselines in Table 14, our approach based on a single classifier compares well to these systems. For example, the best result in the competition (F1 87.21) is achieved by the semi-supervised alternating structure optimization (ASO) method that also uses gene name lexicon, classifier combination, post-processing, etc. The best performance of WCL (F1 87.86) (the last row in Table 14) is achieved by using only basic word attributes, with no gene name dictionaries, and with a single classifier. Similarly, both rank 2 and rank 3 systems in the BioCreativeII competition (F1 86.83 and F1 86.57) include a rich set of features, and rely on multiple-classifier output integration, with bidirectional training of CRF models and a dictionary-based postprocessing for rank 2 system [29].

Using domain dictionaries (e.g., protein/gene name lists) as well as classifier combinations can be easily adopted in our framework to further improve predictive performance. We provide one such example and add a list of known gene names to the system (from NCBI gene list + UNIPROT gene names, ~0.5M entries total). We use the presence of the word in the list as an additional feature for CRF. By adding domain knowledge in this form, we could further improve prediction performance (Table 13). The best state-of-the-art system [30] GM system (to our knowledge) is reported to achieve a slightly higher F1 score of 88.76, however it relies on using very large lexicons (over 15M protein/gene names) with dictionary-based post-processing and multi-classifier setting. The best performance in Table 13 (F1 88.17) is achieved by adding gene name dictionary feature onto the multiple WCL which is slightly higher than the performance of the multiple WCL alone (F1 87.86).

Method	**Baseline (CRF)**	**+Dict.**	**+Dict. +LM+SLLP**
Words+Caps	82.02	**84.26**	**87.41**
All word features	86.34	**87.11**	**88.17**

Table 13: F1 score on the test set for BioCreativeII GM. Adding domain knowledge (gene name dictionary).

6 Conclusion & Discussion

Efficient information retrieval and robust natural language processing on biomedical literature is essential to life science research community. In this work we proposed a novel semi-supervised framework for bioNER through learning compact word representations from a large unlabeled corpus in both semi-supervised and unsupervised settings. In unsupervised setting, LM tries to represent word semantic information through learning natural bio-literature text. While with semi-supervision, SLLP method tries to capture word target label patterns through supervised signals from trained model. Both learned word vector embeddings are fed into a VQ steps for learning codebooks. The resulting codewords are then used to augment features of labeled sets for re-training. We applied this method and several extensions to labeling gene names in bio-literature sentences, where we obtained improvements over the supervised and semi-supervised baselines tested. It yields state-of-the-art performance without the need of complex linguistic features and domain lexicons on BioCreativeII GM task.

WCL is applicable to any other sequence labeling tasks and it could be generalized beyond word-level tagging as well. For instance, a word's embedding could be learned to represent its distribution patterns in natural *documents*. In text categorization problems (e.g., document classification or advertisement recommendation), SLLP of a word could represent the distribution of labels of the *documents* containing that word.

References

[1] CRF++: Yet Another CRF toolkit. http://crfpp.sourceforge.net/.

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT' 98*, pages 92–100, New York, NY, USA, 1998. ACM.

[3] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, 1992.

[4] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*. MIT Press, 2006.

[5] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on EMNLP*, pages 100–110, 1999.

[6] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive SVMs. *J. Mach. Learn. Res.*, 7:1687–1712, 2006.

[7] R. Collobert and J. Weston. A unified architecture for nlp: deep neural networks with multitask learning. In *ICML '08*, pages 160–167, 2008.

[8] H. Daumé III. Cross-task knowledge-constrained self training. In *EMNLP'08*, pages 680–688, Honolulu, Hawaii, October 2008.

[9] G. Druck, G. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *SIGIR '08*, pages 595–602, 2008.

[10] Erik and F. De Meulder. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *HLT-NAACL'03*, pages 142–147, 2003.

Method	**Precision**	**Recall**	**F1**
Baseline 1: BioCreativeII competition (best) [29], semi-supervised method + extensive dictionaries, features	88.48	85.97	87.21
BioCreativeII competition (rank 2), supervised multi-classifier, dictionary	89.30	84.49	86.83
BioCreativeII competition (rank 3), supervised multi-classifier, SVMs+CRFs, rich feature set	84.93	88.28	86.57
Baseline 2: CRF (Words+Caps) supervised	87.84	76.92	82.02
Baseline 3: CRF (Words+Caps) + (unsupervised) co-occurrence	88.52	79.42	83.72
CRF (Words+Caps) + (unsupervised) WCL LM	89.60	83.03	86.19
CRF (Words+Caps) + (semi-supervision) WCL SLLP	89.58	82.93	86.12
CRF (Words+Caps) + multiple WCL (LM + SLLP)	90.12	84.39	87.16
CRF (All word features) + multiple WCL (LM + SLLP)	90.70	85.19	**87.86**

Table 14: A summary of comparison with previous results and baselines on BioCreativeII GM.

[11] A. Gersho and R. M. Gray. *Vector quantization and signal compression*. Kluwer Academic Publishers, Norwell, MA, USA, 1991.

[12] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *NIPS'05*, pages 529–536, Cambridge, MA, 2005. MIT Press.

[13] F. Jiao, S. Wang, C.-H. Lee, R. Greiner, and D. Schuurmans. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *ACL'06*, pages 209–216, 2006.

[14] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML'99*, pages 200–209, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[15] R. J. Kate and R. J. Mooney. Semi-supervised learning for semantic parsing using support vector machines. In *NAACL/HLT'07*, pages 81–84, 2007.

[16] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. GENIA corpus–semantically annotated corpus for biotextmining. *Bioinformatics*, 19 Suppl 1:i180–i182, 2003.

[17] T. Koo and M. Collins. Hidden-variable models for discriminative reranking. In *HLT '05*, pages 507–514, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[18] Z. Kozareva, B. Bonev, and A. Montoyo. Self-training and co-training applied to spanish named entity recognition. In *MICAI'05: Advances in Artificial Intelligence*, 2005.

[19] M. Krallinger, A. Morgan, L. Smith, F. Leitner, L. Tanabe, J. Wilbur, L. Hirschman, and A. Valencia. Evaluation of text-mining systems for biology: overview of the second biocreative community challenge. *Genome Biol*, 9 Suppl 2:S1, 2008.

[20] M. Krallinger, A. Valencia, and L. Hirschman. Linking genes to literature: text mining, information extraction, and retrieval applications for biology. *Genome Biol*, 9 Suppl 2:S8, 2008.

[21] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML'01*, 2001.

[22] G. Mann and A. McCallum. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *ACL'08*, pages 870–878, 2008.

[23] S. Miller, J. Guinness, and A. Zamanian. Name tagging with word clusters and discriminative training. In D. M. Susan Dumais and S. Roukos, editors, *HLT-NAACL'04*, pages 337–342, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

[24] D. Nadeau and S. Sekine. A survey of named entity recognition and classfication. *Linguisticae Investigationes*, 30(1):3–26, 2007.

[25] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2-3):103–134, 2000.

[26] Y. Qi, R. Collobert, P. Kuksa, K. Kavukcuoglu, and J. Weston. Combining labeled and unlabeled data for word-class distribution learning. In *CIKM'09: Proceedings of the eighth ACM Conference on Information and Knowledge Management*. ACM, 2009.

[27] H. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

[28] H. Shan and D. Gildea. Self-training and co-training for semantic role labeling: Primary report. Technical Report TR891, University of Rochester, Comp. Sci. Dept., 2006.

[29] L. Smith, L. Tanabe, R. Ando, et al., and J. W. Wilbur. Overview of biocreative II gene mention recognition. *Genome Biology*, 9(Suppl 2), 2008.

[30] M. Torii, Z. Hu, C. H. Wu, and H. Liu. BioTagger-GM: A Gene/Protein Name Recognition System. *J Am Med Inform Assoc*, 16(2):247–255, 2009.

[31] T. Zhang, F. Damerau, and D. Johnson. Text chunking using regularized winnow. In *ACL '01*, pages 539–546, Morristown, NJ, USA, 2001.

[32] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML'03*, pages 912–919, 2003.